Docket No.: 10013502-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Joubert Berger et al.

Application No.: 09/896,351 | Confirmation No.: 2270

Filed: June 29, 2001 | Art Unit: 2122

For: SYSTEM AND METHOD FOR TRANSFORMING OPERATING SYSTEM AUDIT DATA TO A DESIRED FORMAT | Examiner: C. Kendall

DECLARATION OF JOUBERT BERGER
SUBMITTED UNDER 37 C.F.R. 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

1. My name is Joubert Berger, I am over 21 years of age, and make this declaration based upon my own personal knowledge. All of the statements contained herein are, in all things, true and correct.

2. I am one of the inventors of the invention claimed in the above-identified patent application.

3. Prior to June 12, 2001, I conceived the idea of a system and method for transforming operating system audit data to a desired format as recited in the pending claims of the above-identified patent application. Accordingly, prior to June 12, 2001, I disclosed my invention to my then employer, Hewlett Packard Company.

4. Attached hereto as Exhibit A is a copy of the invention disclosure that I submitted to Hewlett-Packard Company prior to June 12, 2001, for the filing of a patent application. This invention disclosure establishes my conception of the subject matter of the pending claims prior to June 12, 2001.

5. Hewlett-Packard Company considered the invention disclosure that I submitted and approved the filing of a corresponding patent application. The application was filed with the USPTO on June 29, 2001.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: 2/12/04

(Joubert Berger)

Write in Dark Ink on Front Side Only, Please

INVENTION DISCLOSURE

PDNO 3502 DATE RCVD

PAGE ONE OF ___

ATTORNEY LJG/152

Descriptive Title
Generic data transformation for audit event records

Name of Project:

Product Name or Number:

Signature of Inventor(s): Pursuant to my (our) employment agreement, I (we) submit this disclosure on this date:

Employee No.	Name	Signature	Telnet	Mailstop	Entity & Lab Name
	Scott Leerssen				
	Joubert Berger				

(If more than four inventors, include additional information on another copy of this form and attach to this document)

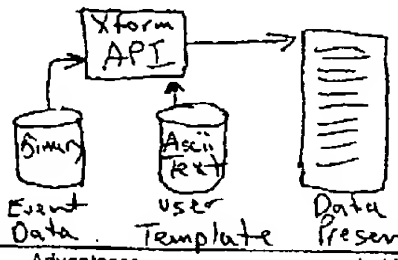
REDACTED

Write in Dark Ink on Front Side Only. Please

Description of Invention:

A. Description of the construction and operation.

This provides a mechanism which can be used to present audit event data in any desired format. An audit event transformation API reads binary audit event data and formats it according to rules in an ASCII text file. The text file is user defined and contains drop points for event data, surrounded by other desired output text.



B. Advantages over what has been done before.

The output format is free-form, and can be fed to any processor including, but not limited to:

- Web browser via HTML
- XML parser
- Event correllation engine for intrusion detection
- Text file
- comma separated value (CSV) file for spreadsheet data

C. Problems solved

This removes the strict, and sometimes arbitrary format of audit event data output.

REDACTED

REDACTED

Functional Specification

Title:	Trusted Linux Audit Functional Specification
Author:	Scott Leerssen
Product & Version:	Trusted Linux Alpha 1
Functional Area:	Trusted Linux
Status:	☐ Not Complete ☒ Draft Ready for Review ☐ Reviewed ☐ Revisions Complete

REDACTED

REDACTED

REDACTED

This document describes the functionality required for an audit mechanism on the Trusted Linux product.

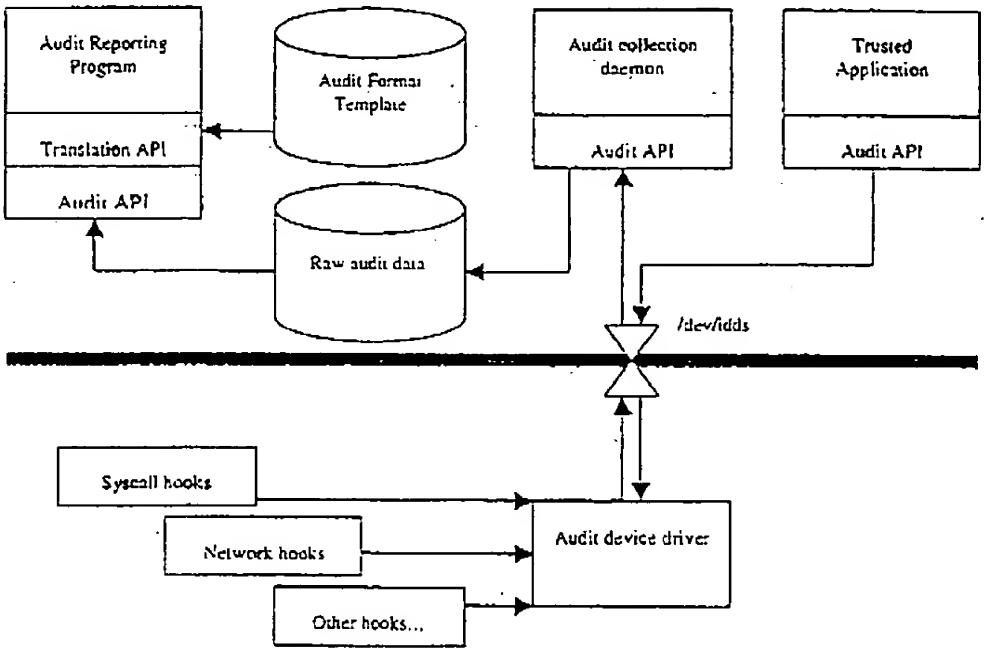


Figure 1

Figure 1 illustrates the interaction of the components of the Trusted Linux audit system. The major components consist of:

- Linux kernel hooks – modifications in the kernel to report audit events.
- Audit device driver – collects audit event data from the kernel and provides an interface for user space applications to collect and report events.
- Audit application programming interface (API) – provides a user level library for reading and writing audit event data to and from the audit device driver
- Audit collection daemon – user space program that collects raw audit event data and writes it to a storage device

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

REDACTED

5.6 Audit Transformation API

Although the heart of the Trusted Linux audit system is the IDDS kernel audit, the audit transformation layer is the where the event data comes to life.

Historically, one of the major inhibitors to the usefulness of audit data has been the format in which it is presented. Usually, the data is either human readable or machine parsable, but rarely both. In some cases, one could argue that audit data falls into neither one of these categories.

This usability issue is where the audit transformation API comes into play. By utilizing a user defined template, the API reads binary audit data and formats it into any desired representation: comma separated lists, XML, HTML, plain text, etc. This list is limited only to a user's imagination. Of course, the binary data could even be streamed in its original format to a collection server for later processing or correlation.

The following sections describe functionality of the audit transformation API library. Although examples are given for each function, they should not be considered design constraints, but merely a guide for determining necessary entry points.

5.6.1 Operational Description

The audit transformation API (ATA) provides a user level library for access to audit data spooled to a device. The API allows a user to define the view in which the audit data should be presented (e.g. XML stream, ASCII stream, etc.).

REDACTED

REDACTED

REDACTED

REDACTED